PTO/SB/21 (04-07)
Approved for use through 09/30/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/600,671
Filing Date	June 20, 2003
First Named Inventor	Carl Staelin
Art Unit	2121
Examiner Name	Michael B. Holmes
Attorney Docket Number	200309618-1
Total Number of Pages in This Submission	

ENCLOSURES (*Check all that apply*)

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) ______________
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☑ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please Identify below):

Remarks

No fee is due for the attached second appeal brief. The $500 fee for was paid with a first appeal brief, which was filed on 11/21/2006.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Law Office of Hugh P. Gortler		
Signature	/Hugh Gortler #33,890/		
Printed name	Hugh P. Gortler		
Date	7/16/2007	Reg. No.	33,890

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name		Date	

Patent
Docket No. 200309618-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

APPEAL NO. ________________

In re Application of:
Carl Staelin et al.

Serial No. 10/600,671
Filed: June 20, 2003

Confirmation No.6065
Group Art Unit: 2121
Examiner Michael B. Holmes

For: NEURAL NETWORK TRAINED WITH SPATIAL ERRORS

APPEAL BRIEF

Hugh P. Gortler, Esq.

Hewlett-Packard Company
Intellectual Property
Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

(949) 454-0898

INDEX

1. <u>REAL PARTY IN INTEREST</u>

The real party in interest is the assignee, Hewlett-Packard Development Company.

2. <u>RELATED APPEALS AND INTERFERENCES</u>

No appeals or interferences are known to have a bearing on the Board's decision in the pending appeal.

3. <u>STATUS OF CLAIMS</u>

Claims 1-37 are pending in this application.

Claims 34-36 are allowed.

Claims 1-33 and 37 are rejected.

The rejections of claims 1-33 and 37 are being appealed.

4. <u>STATUS OF AMENDMENTS</u>

No amendment has been filed subsequent to the final office action, dated July 21, 2006.

5. <u>SUMMARY OF CLAIMED SUBJECT MATTER</u>

Image upscaling typically involves magnifying an entire image or a portion of an image. For example, an image upscaled by a factor of two may have a 2x2 block of pixels corresponding to each pixel in the original image. Pixel values in each 2x2 block of the upscaled image are predicted from pixel values in the original image.

Neural networks can be used to perform image upscaling. As described in paragraph 19 of the specification, a neural network is defined by its nodes, connections, and connection weights. A weight vector is the vector of connection weights between each pair of connected nodes in the neural network. An example of a neural network is illustrated in Figure 5 and described in paragraph 51 of the specification. The exemplary neural network 512 includes input, hidden and output nodes 512-516.

A neural network can be trained to upscale an image. A high resolution image may be downsampled (e.g., by pixel averaging) to produce a corresponding low resolution image. The high resolution image is referred to as a target image. The low resolution image is inputted to the neural network, which produces an upscaled image. Training involves optimizing these weight values so as to reduce the error between the upscaled image and the high resolution image.

Base claim 1 recites a method of training a neural network with input data. The method comprises using the neural network to rescale the input data, determining errors for the rescaled data, and using neighborhoods of the errors to adjust connection weights of the neural network.

An example of training a neural network for image upscaling is illustrated in Figure 3b and described in paragraphs 24-35 of the specification. The neural network is used to upscale the input image (block 314 of Figure 3b and paragraph 26 of the specification). Errors for the upscaled (output) image are determined (block 316 of Figure 3b and paragraph 27 of the specification). For example, an error image may be formed by subtracting the upscaled image from the target image.

Neighborhoods of the errors are used to adjust the connection weights (blocks 318-322 of Figure 3b and paragraphs 28-33 of the specification). An exemplary neighborhood of image pixels is illustrated in Figure 3a. The pixel being processed is the center pixel z_{11}.

Paragraph 28 provides an example of computing derivatives of the errors with respect to the upscaled image (block 318). The derivative for a predicted pixel in the upscaled image is a function of differences between predicted values in a spatial neighborhood of the upscaled image and the corresponding pixel values in the target image. A simple function is described, in which the derivative is the sum of partial derivatives of the neighboring pixels in a spatial neighborhood.

According to paragraph 31, once the full derivatives for the pixels in the rescaled image have been generated, back-propagation is performed to compute error gradients (block 320). According to paragraph 31, the error gradients are used to adjust the node weights to reduce the network errors (block 322).

An example of non-gradient based training is illustrated in Figure 8 and described in paragraph 69. The non-gradient based training also makes use of neighborhoods of errors.

Base claim 19 recites a method of using input data and target data to train a neural network. The method comprises using the neural network to generate predicted values from the input data, determining errors for the predicted values, and back-propagating the errors through the neural network. The error for each predicted value is a function of differences between predicted values in a spatial neighborhood and the corresponding values in the target data.

An example of training a neural network for image upscaling is illustrated in Figure 3b and described in paragraphs 24-35 of the specification. The neural network is used to generate predicted values as illustrated by block 314 of Figure 3b and described in paragraph 26 of the specification. The errors for the predicted values are determined as illustrated by blocks 316-318 of Figure 3b and described in paragraphs 28-30. Back propagation is illustrated by block 320 of Figure 3b and described in paragraphs 31-32.

Base claim 20 recites apparatus for training a neural network on input data. The apparatus comprises means for using the neural network to rescale the input data, means for determining errors for the rescaled data, and means for using neighborhoods of the errors to adjust the connection weights.

An exemplary apparatus is illustrated in Figure 7 and described in paragraphs 80-83 of the specification. A processor 712 and memory 714 provide the means for using the neural network to rescale the input data, determining errors for the rescaled data, and using neighborhoods of the errors to adjust the connection weights. An example of rescaling the input data, determining the errors, and using the errors to adjust connection

weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

Base claim 21 recites apparatus for training a neural network on input data. The apparatus comprises a processor programmed to use the neural network to rescale the input data, determine errors for the rescaled data, and use neighborhoods of the errors to adjust connection weights of the neural network.

An exemplary apparatus is illustrated in Figure 7 and described in paragraphs 80-83 of the specification. The processor is referenced by numeral 712. An example of rescaling the input data, determining the errors, and using the errors to adjust connection weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

Base claim 37 recites an article for causing a processor to use input data to adjust connection weights of a neural network. The article comprises computer memory and data encoded in the computer memory. The data causes the processor to use the neural network to rescale the input data, determine errors for the rescaled data, and use neighborhoods of the errors to adjust the connection weights of the neural network.

An example of the article is illustrated in Figure 7 and described in paragraphs 79-81 of the specification. A processor is referenced by numeral 712. Computer memory 714 is encoded with a program 720 and a neural network 716. An example of rescaling the input data, determining the errors, and using the errors to adjust connection weights is described in paragraphs 26-33 and illustrated by blocks 314-322 of Figure 3b.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-33 and 37 are rejected under 35 USC §101 as reciting nonstatutory subject matter.

7. ARGUMENT

REJECTION OF CLAIMS 1-33 AND 37 UNDER 35 U.S.C. §101 AS RECITING NONSTATUTORY SUBJECT MATTER

The office action dated 7 March 2007 alleges that claims 1-33 and 37 are drawn to an abstract idea rather than a practical application of an abstract idea which would produce a useful, concrete or tangible result. The office action further alleges that the claims fail to recite a practical application and is insufficient to establish a real world practical result.

Claims 1-33 and 37 all recite a neural network, in the preamble, or the body, or both.

A neural network is not an abstract idea. It does not exist in nature. It is not a mental process. It is a man made structure (see paragraphs 61-64 of the specification).

Claims 1-16 and 19 recite methods for transforming this man made structure (e.g., by adjusting connection weights). Claims 20-33 recite apparatus for transforming this man made structure. Claim 18 recites

apparatus including this man made structure, as well as subject matter for transforming it. The transformation is concrete and tangible.

A neural network has a wide range of uses. For example, a neural network can be trained for image upscaling, which typically involves magnifying an entire image or a portion of an image (see paragraphs 1-3 of the specification). Therefore, training a neural network is useful. It has real world value.

Claims 1-33 all recite training a neural network, including adjusting its connection weights. Claim 37 also recites adjusting the connection weights of a neural network. Therefore, claims 1-33 and 37 recite useful subject matter.

There is no legal requirement for <u>claims</u> to recite a useful application. Section 101 requires a claimed invention to be useful. MPEP 2107, which sets forth utility guidelines, states that utility is determined by reading the claims and the written description (see 2107.II.B). According to the specification of the present application, the claimed invention is useful for image processing in general and image upscaling in particular.

Moreover, some of the claims at issue do indeed recite a useful application. Claims 13-16, and 32-33 recite image upscaling and gamut mapping. Claims 2, 9 and 29 recite image processing. Claim 12 recites blurred edges (of an image).

For these reasons, claims 1-33 and 37 recite statutory subject matter. Therefore, the '101 rejection of claims 1-33 and 37 should be withdrawn.

The office action cites the Interim Guidelines for Examination of Patent Applications for Subject Matter Eligibility, but does not follow them. The office action simply cuts and pastes passages from the Guidelines about practical application; results that are useful, concrete and tangible; pre-emption; and safe harbor. However, the office action doesn't apply these passages to specific elements of the claims. It simply makes sweeping generalizations about computer programs. It does not identify and evaluate each claim limitation (as required by MPEP 2106.II.C). It offers no reasons as arguments supporting the conclusion of nonstatutory subject matter. Yet MPEP 2106.1V.B states "The burden is on the USPTO to set forth a *prima facie* case of unpatentability. Therefore if USPTO personnel determine that it is more likely than not that the claimed subject matter falls outside all of the statutory categories, they must provide an explanation." MPEP 2106.1V.D also states that USPTO personnel has the initial burden to identify and explain in the record the reasons why a claim is for an abstract idea with no practical application. The office action has not met this burden.

Moreover, the office action doesn't heed MPEP 2106.II.A, which states "The purpose of this requirement [for useful, concrete and tangible result] is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents

nothing more than an idea or concept, or is simply a starting point for future investigation or research." As discussed above, training a neural network has real world value.

Finally, the office action doesn't heed MPEP 2106.IV.C.2.(1) about transforming a man made structure ("Practical Application by Physical Transformation"). MPEP 2106.IV.C.2.(1) states "USPTO personnel first shall review the claim and determine if it provides a transformation or reduction of an article to a different state or thing. If USPTO personnel find such a transformation or reduction, USPTO personnel shall end the inquiry and find that the claim meets the statutory requirement of 35 U.S.C. 101."

The office action does not follow MPEP 2106.01, which states "functional descriptive material consists of data structures and computer programs which impart functionality when employed as a computer component." Claim 37 recites computer memory encoded with a program that imparts functionality to a processor (the program causes a processor to use a neural network to rescale the input data; determine errors for the rescaled data; and use neighborhoods of the errors to adjust the connection weights of the neural network). Thus, the subject matter of claim 37 is considered statutory under MPEP 2106.01.

For the reasons above, the rejections of claims 1-33 and 37 should be withdrawn. The Honorable Board of Patent Appeals and Interferences is respectfully requested to reverse these rejections.

Respectfully submitted,

/Hugh Gortler #33,890/
Hugh P. Gortler, Esq.
Registration No. 33, 890

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

(949) 454-0898

Date: July 12, 2007

8. <u>CLAIMS APPENDIX</u>

1. (Original) A method of training a neural network with input data, the neural network including a plurality of connection weights, the method comprising:

using the neural network to rescale the input data;

determining errors for the rescaled data; and

using neighborhoods of the errors to adjust the connection weights.

2. (Original) The method of claim 1, wherein the input data represents a set of images, and wherein the neighborhoods are spatial error neighborhoods.

3. (Original) The method of claim 1, wherein the error neighborhoods are used with a non-gradient algorithm to adjust the connection weights.

4. (Original) The method of claim 1, wherein the error neighborhoods are used to generate derivatives of total error with respect to a neighborhood of errors; wherein gradients are computed from the derivatives; and wherein the gradients are used to adjust the connection weights.

5. (Original) The method of claim 4, wherein each derivative is computed as the sum of the partial derivatives of the errors in an error neighborhood.

6. (Original) The method of claim 4, wherein each derivative of total error with respect to a neighborhood of errors is proportional to a product of a penalty matrix and an error vector, the error vector describing the neighborhood of errors, the penalty matrix punishing any spatially correlated errors.

7. (Original) The method of claim 6, wherein the penalty matrix is positive definite, and includes weights that penalize undesirable patterns of errors.

8. (Original) The method of claim 6, wherein the penalty matrix is based on use of a pattern detector that detects the spatially correlated errors.

9. (Original) The method of claim 1, wherein determining the errors includes forming an error image from the rescaled data, identifying patterns in the error image, and punishing the spatially correlated errors in the error image.

10. (Original) The method of claim 1, wherein input and output data of the neural network are coded to improve the neural network accuracy.

11. (Original) The method of claim 1, wherein the errors are a combination of SSE and spatial errors.

12. (Original) The method of claim 11, wherein SSE is applied to crisp edges and spatial errors are applied to blurred edges.

13. (Original) A method of upscaling an input image, the method comprising using the neural network trained according to claim 1.

14. (Original) The method of claim 13, wherein the input and upscaled images are color images; wherein the input image is upscaled by pixel replication; a luminance channel of the input image is upscaled by the neural network; and the upscaled luminance channel and the pixel-replicated image are used to generate the upscaled color image.

15. (Original) The method of claim 14, wherein the using the upscaled luminance channel and the pixel-replicated image include adding deltas to pixels in the pixel-replicated image, each delta computed as the difference between the corresponding luminance value in the upscaled luminance channel and the corresponding luminance value in the input luminance channel.

16. (Original) The method of claim 15, wherein using the upscaled luminance channel and the pixel-replicated image further includes gamut mapping the upscaled image.

17. (Original) An article comprising computer memory encoded with data upscaled by the neural network trained according to the method of claim 1.

18. (Original) Apparatus comprising a processor programmed with a neural network, the network trained according to the method of claim 1.

19. (Original) A method of using input data and target data to train a neural network, during training, the method comprising:

using the neural network to generate predicted values from the input data;

determining errors for the predicted values, the error for each predicted value a function of differences between predicted values in a spatial neighborhood and the corresponding values in the target data; and

back-propagating the errors through the neural network.

20. (Original) Apparatus for training a neural network on input data, the apparatus comprising:

means for using the neural network to rescale the input data;

means for determining errors for the rescaled data; and

means for using neighborhoods of the errors to adjust the connection weights.

21. (Original) Apparatus for training a neural network on input data, the neural network having a plurality of connection weights, the apparatus comprising a processor programmed to use the neural network to rescale the input data; determine errors for the rescaled data; and use neighborhoods of the errors to adjust the connection weights of the neural network.

22. (Original) The apparatus of claim 21, wherein the input data represents images, and wherein the neighborhoods are spatial error neighborhoods.

23. (Original) The apparatus of claim 21, wherein the processor is programmed to use the error neighborhoods and a non-gradient algorithm to adjust the connection weights.

24. (Original) The apparatus of claim 21, wherein the error neighborhoods are used to generate derivatives of total error with respect to a neighborhood of errors; wherein gradients are computed from the derivatives; and wherein the gradients are used to adjust the connection weights.

25. (Original) The apparatus of claim 24, wherein each derivative is computed as the sum of the partial derivatives of the errors in an error neighborhood.

26. (Original) The apparatus of claim 24, wherein each derivative of total error with respect to a neighborhood of errors is proportional to a product of a penalty matrix and an error vector, the error vector describing the neighborhood of errors, the penalty matrix punishing any spatially correlated errors.

27. (Original) The apparatus of claim 26, wherein the penalty matrix is positive definite, and includes weights that penalize undesirable patterns of errors.

28. (Original) The apparatus of claim 26, wherein the penalty matrix is based on use of a pattern detector that detects the spatially correlated errors.

29. (Original) The apparatus of claim 21, wherein determining the errors includes forming an error image from the rescaled data, identifying patterns in the error image, and punishing the spatially correlated errors in the error image.

30. (Original) The apparatus of claim 21, wherein the processor is programmed to code input and output data of the neural network to improve the neural network accuracy.

31. (Original) The apparatus of claim 21, wherein the errors are a combination of SSE and spatial errors.

32. (Original) The apparatus of claim 21, wherein the input and upscaled images are color images; wherein the processor is programmed to upscale the input image by pixel replication, use the neural network to upscale a luminance channel of the input image; and generate the upscaled color image from the upscaled luminance channel and the pixel-replicated image.

33. (Original) The apparatus of claim 32, wherein the processor is further programmed to perform gamut mapping of the upscaled image.

34. (Original) Apparatus for rescaling a color image, the apparatus comprising:

means for rescaling the input image by pixel replication;

a neural network that has been trained to rescale a luminance channel of the color image, the neural network for producing a rescaled luminance image; and

means for using the rescaled luminance image and the pixel-replicated image to generate a rescaled color image.

35. (Original) The apparatus of claim 34, wherein the use of the rescaled luminance image and the pixel-replicated image includes adding deltas to pixels in the pixel-replicated image, each delta computed as the difference between the corresponding luminance value in the rescaled luminance image and the corresponding luminance value in the input luminance channel.

36. (Original) The apparatus of claim 32, further comprising means for gamut mapping the rescaled color image.

37. (Original) An article for causing a processor to use input data to adjust connection weights of a neural network, the article comprising:

computer memory:

data encoded in the computer memory, the data causing the processor to use the neural network to rescale the input data; determine errors for the rescaled data; and use neighborhoods of the errors to adjust the connection weights of the neural network.

9. EVIDENCE APPENDIX

None

10. RELATED PROCEEDINGS APPENDIX

None